

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE
ZNANOSTI
Ak. god. 2014./2015.

Dario Poljak

Prilagodljivi web dizajn u kontekstu mobilnog weba

Završni rad

Mentor: dr. sc. Kristina Kocijan, doc.

Zagreb, 2015.

Sadržaj:

Sadržaj:.....	1
1. Uvod	2
1.1. Osnovni pojmovi	3
1.1.1. HTML.....	3
1.1.2. CSS.....	3
1.1.3. JavaScript	3
1.1.4. PHP.....	3
1.1.5. <i>User-agent</i>	4
1.1.6. <i>Framework</i>	4
1.2. Mobilni uređaji	4
1.2.1. Mobiteli	4
1.2.2. Tableti.....	5
1.2.3. Netbook računala.....	5
2. Povijest mobilnog weba.....	6
2.1. WAP	6
2.2. <i>i-mode</i>	7
3. Rješenja za izradu stranica mobilnog weba.....	9
3.1. Preusmjeravanje	9
3.1.1. Način preusmjeravanja	10
3.1.2. <i>Pluginovi</i>	10
3.1.3. Posebne skripte.....	11
3.1.4. Problemi u pristupu	12
3.2. Prilagodljivi web dizajn.....	15
3.2.1. Bootstrap	16
3.2.2. Ostali frameworkci	23
4. Zaključak	25
5. Popis literature	26

1. Uvod

Odmah na početku treba odrediti na što točno se odnosi pojam mobilni web jer je njegovim razvojem došlo do podjele u dvije struje. Mobilni web kao zaseban web razvijan i odvojen od ostatka weba te jedinstveni web koji se na drukčiji način prikazuje na mobilnim uređajima (Hay, 2011.). U ovom će se radu termin 'mobilni web' koristiti u njegovom drugom značenju, dakle kao jedinstveni web koji se na drukčiji način prikazuje na mobilnim uređajima te je iz tog razloga potrebna prilagodba njegova sadržaja, što je i tema ovoga rada.

U prvom dijelu rada ću objasniti osnovnu terminologiju koja se veže uz web općenito i navesti mobilne uređaje koji se danas koriste za prikaz web stranica. Potom ću proći kroz povijest i razvitak mobilnog weba te spomenuti probleme i rješenja koje je njegov razvoj donio počevši od WAP protokola i prikaza šturih te krajnje jednostavnih stranica na malenim ekranima prvih mobilnih uređaja koji su imali sposobnost spajanja na Internet preko preusmjeravanja mobilnih korisnika na takozvane „m.“ stranice koje su posebno dizajnirane, s mobilnim uređajima u vidu do stranica i aplikacija s prilagodljivim dizajnom ciljajući posebno pametne mobitele i tablete.

Cilj ovog rada je usporediti postojeće pristupe izradi mobilnih web stranica i njihovom prikazu te obraditi razlike u radu istih. Na kraju namjeravam dati prijedlog pristupa kojim se najveći broj web korisnika može učiniti zadovoljnim i ostati vjeran ideji W3C-a o jedinstvenom iskustvu na webu bez obzira na uređaj s kojim se webu pristupa (Mills, 2011.).

1.1. Osnovni pojmovi

U ovom poglavlju namjeravam obraditi osnovnu terminologiju koju ću koristiti kroz rad.

1.1.1. HTML

HTML je skraćenica za *Hyper Text Markup Language*. Radi se o jeziku za označavanje kojim se opisuju ili izrađuju web stranice. Sastoji se od html oznaka koje mogu uključivati attribute i njihove vrijednosti i sadržaja koji se prikazuje na stranici. Oznake definiraju elemente, atributi detaljnije opisuju oznake, a sadržaj je ono što se nalazi unutar oznaka i na koji one utječu. Sintaksu u html-u definiraju oznake koje se moraju nalaziti unutar znakova manje (<) i veće (>) te svaki otvoreni element mora biti i zatvoren na način da se ponovi, samo se na kraju unutar oznake dodaje kosa crta unatrag (\)(W3Schools, 2015.).

1.1.2. CSS

CSS je skraćenica za *Cascading style sheet* a radi se o dokumentu u kojem se definira vizualni izgled HTML stranice. Najčešće se radi o samo jednom dokumentu koji se primjenjuje nad cijelim web sjedištem. Osnovna funkcionalnost css-a je u činjenici da se set pravila za prikaz primjenjuje nad svim elementima istog imena. Pomoću toga možemo na sve elemente <p> na web sjedištu primijeniti svojstva poravnavanja i boje slova bez da to moramo pisati za svako ponavljanje tog elementa. Jednako funkcionira i s klasama koje možemo navesti unutar oznaka uz ključnu riječ *class*. Klasi se dodjeljuje proizvoljno ime i onda se nad svim elementima koji dijele ime klase primjenjuje isti set pravila koji css određuje (W3Schools, 2015.).

1.1.3. JavaScript

Osmislio ga je 1995. Brendan Eich i postao je standard 1997. JavaScript je skriptni jezik koji se najviše koristi kod izrade web stranica i ujedno je najpopularniji skriptni jezik na svijetu. Izvodi se s klijentske strane, odnosno na uređaju koji otvara web stranicu. Po definiciji JavaScript može biti unutarnji i vanjski. Razlika je u tome što je unutarnji napisan unutar html dokumenta u kojem se izvodi, a vanjski je napisan u posebnom dokumentu koji se zatim poziva na sličan način kao i css datoteka (W3Schools 2015.).

1.1.4. PHP

PHP je skriptni jezik poput JavaScripte, no on se izvodi na poslužitelju i stoga nije ovisan o tome što podržava uređaj na kojem se web stranica gleda. Zbog te svoje karakteristike da se

jednako izvodi bez obzira o uređaju koristi se u mnogim aspektima izrade web stranica poput sustava za kontrolu sadržaja, sustava za učenje na daljinu i mnogih drugih (W3Schools 2015.).

1.1.5. *User-agent*

Radi se o nizu znakova koje internetski preglednik šalje poslužitelju pri posjećivanju web stranice u kojem se nalaze podaci o samom pregledniku, operativnom sustavu i razlučivosti ekrana uređaja na kojem se stranica prikazuje. Na temelju tih podataka poslužitelji mogu prikazati stranicu krojenu za specifični preglednik (Microsoft, 2015.).

1.1.6. *Framework*

U okviru ovog rada *framework* se odnosi na programski *framework* odnosno, skup koda koji je unaprijed napisan u svrhu skraćivanja procesa izrade funkcionalnog programskog rješenja. *Framework* u sebi sadrži unaprijed definirane klase i funkcije kojima se znatno olakšava i ubrzava rad programera koji ne moraju svaki puta iznova pisati rješenja za česte probleme. (Christensson, 2015.).

1.2. Mobilni uređaji

Mobilni web je krovni termin kojim pokrivamo surfanje Internetom u pokretu s bilo kojeg prijenosnog uređaja koji ima pristup Internetu. Međutim, potrebno je definirati na što se sve termin mobilni uređaj odnosi.

Firtman definira mobilni uređaj kao „*prijenosan, osoban, uz vas velik dio vremena, lagan i brz za korištenje, posjeduje neku vrstu mrežne spojivosti*“¹ (Firtman 2013:7). Uz te kriterije možemo napraviti sljedeću podjelu: mobiteli, tableti i netbook računala. Ukratko ću opisati svaki.

1.2.1. Mobiteli

Uređaji čija je primarna namjena zvati i slati poruke. Napretkom tehnologija i dodavanja novih funkcionalnosti došlo je do unutarnje podjele na *feature phones* i pametne telefone.

¹ Prijevod autora. Tekst u originalu glasi: “For the purposes of this book, a mobile device has the following features: it’s portable, it’s personal, it’s with you almost all the time, it’s easy and fast to use, it has some kind of network connection.”

Pri izradi mobilnih web stranica, one se najčešće prilagođavaju pametnim mobitelima jer ljudi, osim osobnim računalima, najčešće webu pristupaju preko mobitela (Murtaugh, 2010.). Stoga će se i ovaj rad najviše baviti prikazom web stranica na mobitelima i usporedbom s njihovom verzijom za osobna računala.

1.2.2. Tableti

Tablet je ravan uređaj koji obično ne stane u džep i ima ekran na dodir dijagonale 7 do 11 inča. Neki koriste mobilne operativne sustave (poput iOS-a umjesto Mac OSX-a), a neki koriste *desktop* operativne sustave optimizirane za ekrane na dodir (na primjer *Windows for tablets* umjesto *Windows Phone*). Tableti imaju istu funkcionalnost kao i pametni telefoni, uključujući *multitasking* operativni sustav, instaliranje aplikacija i potpuni Internet preglednik. Svaki tablet ima podršku za WiFi, a neki i 3G odnosno 4G spojitivost (Firtman, 2013.).

1.2.3. Netbook računala

Netbook računala imaju ekran dijagonale najmanje 9 inča i više su u srodstvu s punokrvnim računalima nego mobilnim uređajima. Neki imaju *desktop* operativne sustave, dok neki imaju pojednostavljene operativne sustave s naglaskom na web (poput *Chromebookova* koji imaju Chrome OS) (Firtman, 2013.).

2. Povijest mobilnog weba

Povezivost mobilnih uređaja s Internetom pojavila se pod kraj 90-ih godina 20. stoljeća. Prema Vodafoneu prvi uređaji sposobni za surfanje WAP stranicama na Internetu u Europi su bili izdani 1999. godine (Vodafone, 2013.). Osim i-mode usluge² koja je bila dominantna u Japanu i još nekim zemljama, vodeći standard na tržištu mobilnog weba bio je WAP (Mills, 2011.).

2.1. WAP

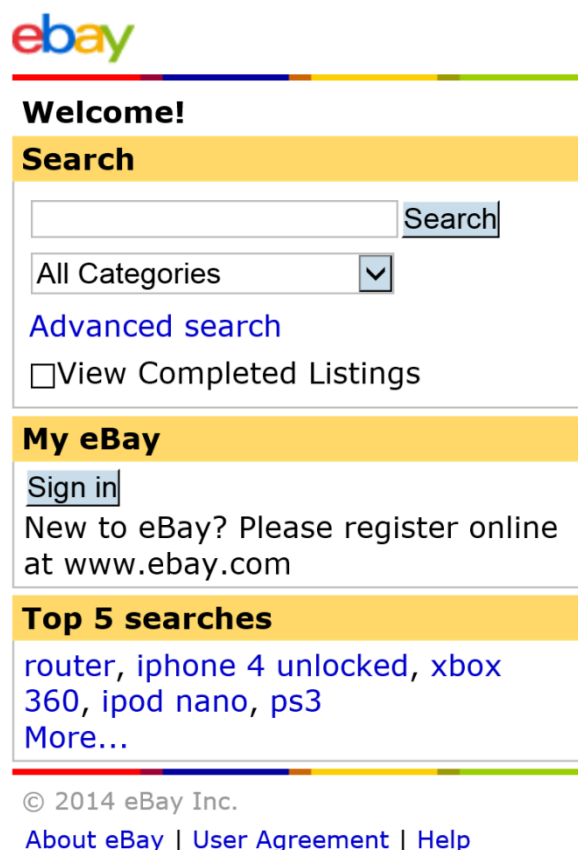
WAP je kratica za *Wireless Application Protocol*, a predstavlja siguran sustav koji dozvoljava korisnicima da pristupaju informacijama trenutno putem prijenosnih uređaja poput mobitela, planera, radija, pametnih telefona i sličnih (Beal, 2012.).

Actually Phone.com, Ericsson, Nokia i mnogi drugi počeli su neovisno jedni o drugima razvijati svoje vlastite standarde, ali uskoro su shvatili da bi puno smislenije bilo usredotočiti razvoj na zajednički standard. Tako je rođen WAP forum, sa željom da se utemelji zajednički format za prijenos Interneta na mobilnim uređajima bez potrebe da se sadržaj mora uređivati posebno za svaku vrstu prijenosnog uređaja (Dhawan, 2001b).

S obzirom da se radi o samim počecima rada na prikazivanju mobilnog weba na uređajima koji su u vidu procesorske snage i dostupne memorije iznimno ograničeni, vizualni identitet stranice se stavljao po strani i na prvom mjestu je bila njena uporabljivost kao što je to vidljivo na slici 1. Prikazana je popularna stranica za Internet kupovinu eBay i uz minimalno sučelje je postignuta njena uporabljivost.

Iako WAP podržava HTML i XML, posebno za potrebe malih ekrana i navigacije jednom rukom bez tipkovnice, primjenom XML jezika nastao je WML jezik. WML može skalirati od dvo-linijskih displeja do grafičkih ekrana poput onih na pametnim telefonima i komunikatorima. WAP također podržava i WML Script koji je sličan JavaScripti, ali ima minimalne zahtjeve za memorijom i procesorskom snagom jer ne sadrži mnoge od nepotrebnih funkcija nađenih u drugim skriptnim jezicima (Beal, 2012.).

² Usluga pristupanja posebno prilagođenim web stranicama putem poslužitelja mobilnog operatera



Slika 1 Popularna stranica za online trgovinu eBay u WAP izdanju

WAP svakako predstavlja početak razvoja mobilnog weba, ali proći će još dosta vremena i nekoliko iteracija dok ne dostigne razinu na kojoj je mobilni web danas.

2.2. *i-mode*

Prvo treba naglasiti da postoji česta greška u usporedbi *i-mode* usluge i WAP standarda. Kao što je ranije navedeno, *i-mode* je plaćena usluga, a prvi ju je uveo japanski teleoperater NTTDoCoMo. Usluga se sastoji od 1 100 službeno odobrenih stranica i otprilike 24 000 neslužbenih stranica kojima korisnici mogu pristupiti uz plaćanje naknade teleoperateru (Dhawan, 2001a).

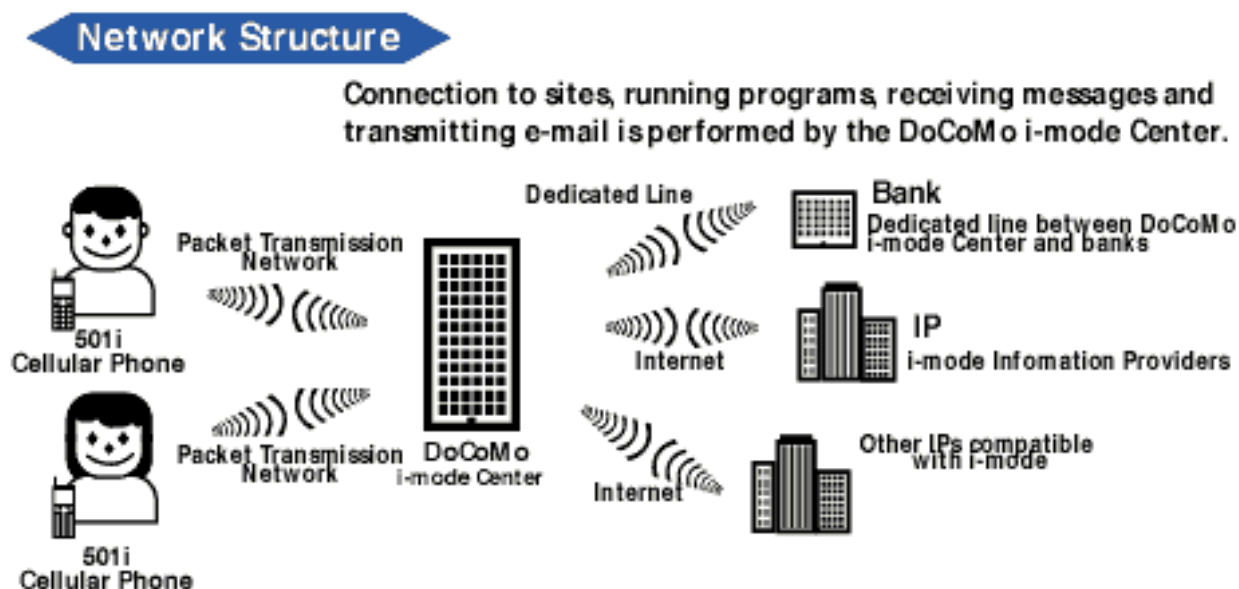
Stranice su posebno prilagođene mobilnim uređajima i za razliku od stranica pisanim za WAP standard, *i-mode* stranice su pisane u *c-HTMLu* odnosno *compact HTML* jeziku (Mills, 2011.).

i-mode funkcioniра na način da se korisnik svojim mobilnim uređajem spaja na poslužitelj svog mobilnog operatera i zatim taj poslužitelj dohvaća željenu *i-mode* stranicu te je šalje na

mobilni uređaj putem posebno razvijenog PDC-P (en. *Packet data communications systems*) protokola.

Usluga *i-mode* se ne temelji toliko na pristupu bilo kojoj stranici na Internetu već na velikoj ponudi stranica koje zadovoljavaju jednostavne potrebe korisnika od provjere voznog reda vlakova i prognoze do slanja e-pošte i razgovaranja s drugim korisnicima (NTTDoCoMo, 2010.). Kao usluga *i-mode* je bila odlično prihvaćena u Japanu, a pojavila se i u nekoliko drugih zemalja pod drukčijim imenom. Uz dozvolu NTT DoCoMo teleoperatera osnovan je tzv. *i-mode* savez koji se najbolje utemeljio u Europi. Prva zemlja nakon Japana koja je prigrlila *i-mode* je bila Njemačka u ožujku 2002. godine gdje je uslugu pružio teleoperater E-Plus, a nakon nje iste godine su uslugu uvele i Nizozemska, Belgija i Francuska. U 2003. godini pridružile su im se Španjolska i Italija. Van Europe je *i-mode* uveden u Tajvanu i nekim predjelima SAD-a (NTTDoCoMo, 2004.).

Uzevši sve u obzir *i-mode* je dobro implementiran skup usluga koji se održao jer nudi točno ono što korisnicima treba, ali gledano iz perspektive ovog rada je vrlo loše rješenje jer odmah na početku dijeli web na mobilne uređaje i računala. Cijeli pristup *i-mode* usluzi se vrši preko teleoperatera i stoga je dostupna samo vrlo ograničenom broju ljudi (Dhawan, 2001a).



Slika 2: Prikaz kako funkcionira pružanje i-mode usluge³

³ Preuzeto sa: <http://www.mobileinfo.com/imode/components.htm>

3. Rješenja za izradu stranica mobilnog weba

Pojavom pametnih telefona koji su stalno povezani s Internetom i dolaze s već unaprijed instaliranim web preglednikom javila se potreba prilagoditi sadržaj na webu mobilnim uređajima. U početku nije bilo strogo definirano kako bi to trebalo napraviti i mnoge stranice su ostale jednostavno onakve kakve jesu, bez ikakve prilagodbe mobilnim uređajima (Everts, 2013.). Takve stranice postoje još dan danas i jedan primjer je na lijevoj strani slike 3.



Slika 3: Primjer stranice bez prilagodbe prikazane na mobilnom uređaju (lijevo) i ista stranica prikazana na *desktop* računalu (desno)

3.1. Preusmjeravanje

Najbrže rješenje za novonastalu situaciju bilo je preusmjeravanje. U suštini, radi se o prikazu jedne stranice korisnicima na osobnim računalima, a sasvim druge korisnicima na mobilnim uređajima. Česta praksa za te stranice je da prije svog naziva imaju „m.“ (npr. www.m.facebook.com) vidi sliku 4.



Slika 4: Dijagram koji prikazuje odvojene stranice za računala i mobilne uređaje⁴

3.1.1. Način preusmjeravanja

Osnovna ideja iza preusmjeravanja vrlo je jednostavna. Ukoliko korisnik pristupa stranici mobitelom onda ga ona preusmjerava na mobilnu verziju, a ukoliko dolazi računalom onda ostaje na stranici namijenjenoj prikazu na računalu.

Problem je u tome što kada ljudi posjećuju vašu web stranicu, njihov preglednik ne zna „Jesam li ja mobilni uređaj?“ ili „Jesam li ja osobno računalo?“. Umjesto toga, preglednik se predstavlja dajući svoj „*user-agent*“, koji u sebi sadrži ime preglednika, verziju i druge informacije o vašem operativnom sustavu. Kako biste preusmjerili mobilnog korisnika, morate odrediti je li njihov preglednik na mobilnom uređaju ili ne. Možete odrediti koji preglednik posjetitelj koristi. Isto tako možete odrediti i druge stvari poput rezolucije njihova ekrana i prihvaćaju li datoteke koje su česte na mobilnim uređajima (Markle, 2015.).

Te informacije se mogu provjeriti na nekoliko različitih načina navođenjem, recimo, posebnih skripta ili *pluginova*.

3.1.2. Pluginovi

Pri izradi web stranica možemo se koristiti i sustavom za upravljanje sadržajem (en. *Web Content Management System* – *WCMS*). *WCMS* korisniku nudi mogućnost za jednostavno uređivanje web stranice i stavljanja sadržaja na nju. *WCMS* se često puta nudi u sklopu

⁴ Preuzeto sa <https://developers.google.com/webmasters/mobile-sites/mobile-seo/configurations/separate-urls?hl=en> (22.5.2015.)

velikih CMS sustava za tvrtke, ali se može pronaći i zasebno. Neki od popularnijih WCMS sustava su Wordpress, Joomla, DotNetNuke i Endplay. (Johnston, 2011.)

Ukoliko se koristimo gotovim rješenjem za izradu web stranica, najvjerojatnije već postoji i gotov dodatak za mobilne web stranice. Međutim, tu nastaje i problem jer svaki WCMS i gotovo rješenje imaju svoje pluginove. Neki od WCMS sustava su Wordpress, Joomla, ModX, TextPattern. Wordpress koristi *WPtouch*, *iThemes*, *Any Mobile ThemeSwitcher* (Wilson, 2014.), dok Django koristi *Django Mobile*, *DjangoMobler*, *DjangoMobility* (Greenfield, Audrey, 2013.).

3.1.3. Posebne skripte

JavaScriptom, PHP-om ili čak CSS-om moguće je provjeriti pristupa li posjetitelj s mobilnog uređaja. U sljedećih nekoliko blokova koda biti će navedene najosnovnije skripte kako bi se usporedio njihov pristup i njihova implementacija u web stranicu.

```
<script type="text/javascript">
  <!--
  if (screen.width <= 800) {
    window.location = "http://m.domain.com";
  }
  //-->
</script>
```

Kodni blok 1 JavaScript kod za preusmjeravanje mobilnih uređaja⁵

Kao što je vidljivo u kodnom bloku 1, JavaScript putem jednostavne logičke operacije provjerava je li širina ekrana manja ili jednaka 800. Ukoliko provjera rezultira istinom putem funkcije `window.location` korisnika se prebacuje na „`http://m.domain.com`“ odnosno na mobilnu verziju web sjedišta.

Slična stvar se može napraviti i provjerom *user-agenta* kojim se preglednik predstavlja pri dolasku posjetitelja na stranicu. Izvadak koda koji provjerava *user-agent* preglednika preko JavaScripta je priložen u kodnom bloku 2.

⁵Preuzeto sa:<http://www.inmotionhosting.com/support/website/redirects/mobile-redirect> (21.5.2015.)

```
<script type="text/javascript">
<!--
if ((navigator.userAgent.match(/iPhone/i)) ||
(navigator.userAgent.match(/iPod/i))) {
    location.replace("http://YOUR-MOBILE-SITE.com");
}
-->
</script>
```

Kodni blok 2 JavaScript kod za preusmjeravanje mobilnih uređaja na temelju user-agenta⁶

```
<link rel="stylesheet" href="screen.css" media="screen"/>
<link rel="stylesheet" href="handheld.css" media="handheld"/>
```

Kodni blok 3 CSS kod za promjenu ukupnog CSS-a na stranici

Kao što je vidljivo iz kodnog bloka 3, CSS koristi nove *media* upite koji su uvedeni u CSS verziji 2, a dovršeni u verziji 3 (W3Schools, 2015.). Mobilni i tableti tipično traže jednu od dvije CSS vrijednosti atributa @media – „screen“ i „handheld“. S obzirom da ne znamo koji će naš korisnik trebati potrebno je implementirati oba (Sandu, 2011.).

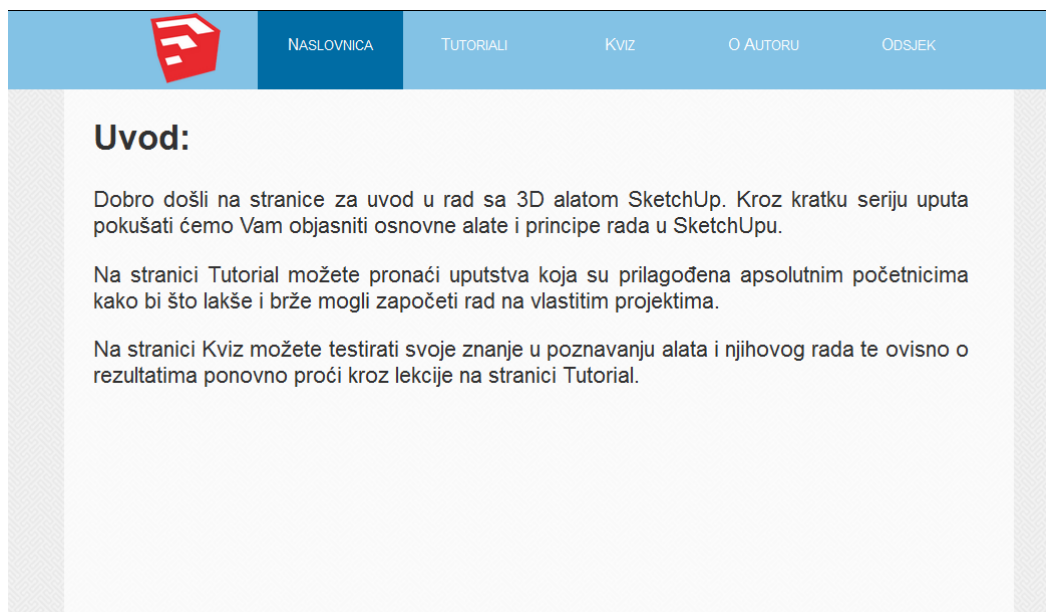
PHP ima pristup vrlo sličan pristupu JavaScripta za provjeru *user-agenta*. Jedina razlika je u sintaksi i tome što u slučaju PHP-a sve obavlja poslužitelj, dok se JavaScript izvodi na uređaju klijenta. Zbog toga se pomoću PHP-a može zaobići problem gdje uređaj kojim klijent pristupa ne podržava JavaScript.

3.1.4. Problemi u pristupu

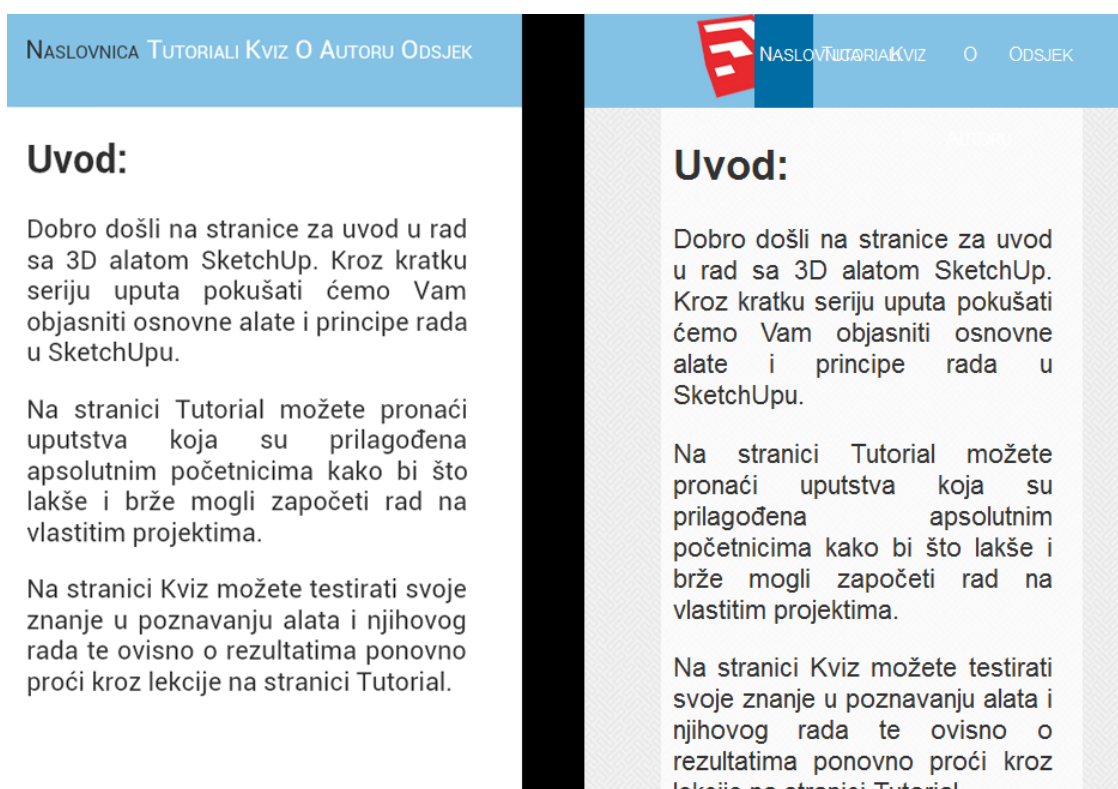
Unatoč tome što se dvije odvojene stranice za mobilne uređaje i računala čine kao idealno rješenje, one imaju velik broj problema s kojima sam se pri izradi web stranica i sam susreo.

Pri korištenju *media* upita velik je problem što ih netbook računala i iznenađujuće velik broj drugih uređaja ne prepoznaje te se stranice ne prikazuju pravilno. Stranica u svojem originalnom stanju na osobnom računalu izgleda kao prikaz na slici 5.

⁶Preuzeto sa: <http://www.designyourway.net/blog/resources/detecting-and-redirecting-mobile-users/> (30.5.2015)



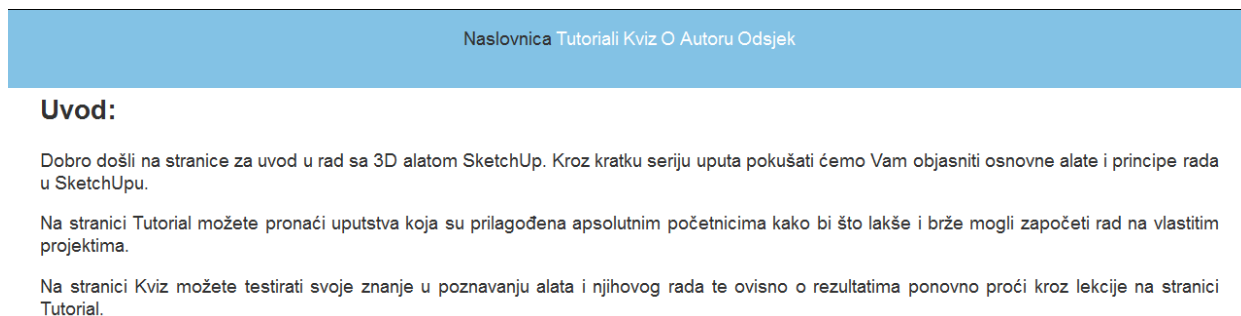
Slika 5 Prikaz izgleda stranice na osobnom računalu



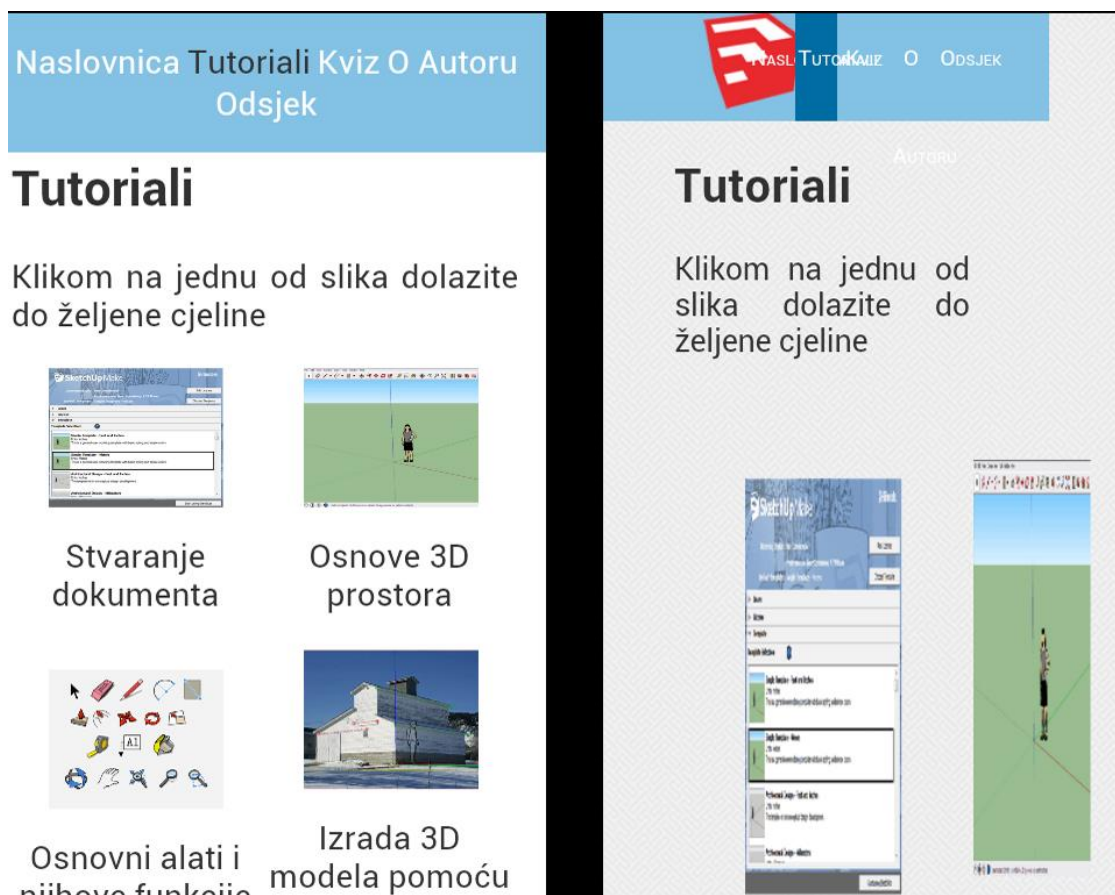
Slika 6 Usporedba kako bi CSS verzija trebala izgledati (lijevo) i kako izgleda u slučaju greške (desno)

U slučaju da se *media* upit ne provede, a stranica se svejedno pokuša prikazati na ekranu niske razlučivosti, dolazimo do nečeg sličnog desnom prikazu na slici 6. Uređaj će pokušati prikazati cijelu stranicu, a s obzirom da ista koristi postotke kako bi elementi što bolje skalirali, sadržaj je uspješno smanjen, no velik je dio navigacijske trake postao beskoristan.

S druge strane nalazi se PHP način preusmjeravanja stranice. Osim što je iznimno zamorno i vremenski zahtjevno konstantno ažurirati listu *user-agenata*, puno puta dolazi i do problema kada korisnik pokušava pristupiti određenom dijelu stranice na kojem se ne izvodi provjera *user-agenata* te se korisniku servira stranica za *desktop* ili mobilni uređaj bez obzira na to što on koristi. Takva situacija je prikazana na slikama 7 i 8.



Slika 7 Mobilna stranica greškom prikazana na ekranu visoke razlučivosti



Slika 8 Ispravan prikaz stranice (lijevo), neispravan prikaz bez preusmjeravanja PHP-a (desno)

Osim što treba predvidjeti jako mnogo uređaja i različitih razlučivosti ekrana, postoji i problem održavanja dviju odvojenih stranica ili barem dva odvojena CSS dokumenta.

U slučaju dvije odvojene stranice svaku promjenu treba obaviti dva puta, jer neka promjena na glavnoj stranici neće biti vidljiva na mobilnoj ako se i na njoj samoj ta ista stvar ne promijeni. Zbog toga češće nego ne, m. stranica je ili naknadna ideja ili hobi projekt sa strane, dok bi zapravo mobilna stranica trebala dobivati pažnje koliko i glavna stranica (Everts, 2013.).

3.2. Prilagodljivi web dizajn

Prilagodljivi web dizajn dozvoljava vam da imate jednu web stranicu koja se automatski prilagođava veličini ekrana na uređaju na kojem se gleda. To se postiže prilagodbom sadržaja, dizajna, navigacije i metoda interakcije kako bi se dostavila ista ugodnost i iskoristivost stranice korisniku na mobilnom uređaju i na osobnom računalu (Ghazarian, 2014.).

Prilagodljivi web dizajn je zapravo ono na što W3C cilja od svog osnutka, a to je jedinstveni web bez podjele na mobilne uređaje, računala, televizije i slično (Mills, 2011.). Prilagodljivi dizajn to dozvoljava jer implementacijom stupaca kao gradivnih elemenata omogućuje da se stranica dinamično skalira na svim uređajima.



Slika 9 Primjer prilagodljivo dizajnirane stranice (<http://stephencaver.com/>) u izvedbi za velike ekrane (desno), srednje (sredina) i male (lijevo)⁷

⁷ Preuzeto sa: <http://designmodo.com/responsive-design-examples/> (2.6.2015.)

Kao što je vidljivo na slici 9, prilagodljivo dizajnirana stranica prilagođava se na temelju veličine ekrana na kojemu je prikazana, putem media upita koji kao mjeru uzimaju širinu prozora preglednika i temelje se na max-width i min-width atributima unutar samog CSS-a (Spurlock, 2013.).

Moguće je pisati vlastita rješenja za stranicu prilagodljiva dizajna, pogotovo ako se radi o izradi na neki način posebne stranice za koju gotova rješenja (iz nekog razloga) ne funkcioniraju na željeni način. Ipak, postoji velika količina gotovih *frameworkova* koji se mogu rabiti ako projekt ne iziskuje poseban dizajn, a čije korištenje olakšava i ubrzava cijeli proces izrade stranice.

Poučen vlastitim iskustvom, zaključio sam kako je isplativije osloniti se na već postojeća rješenja budući su mnoge stvari već unaprijed riješene i nije potrebno gubiti vrijeme na pronalaženje vlastitih rješenja. Na kraju izrade mobilnog prikaza za svoju web stranicu okrenuo sam se *frameworku Bootstrap* koji ću stoga malo detaljnije opisati i potkrijepiti vlastitim primjerima.

3.2.1. Bootstrap

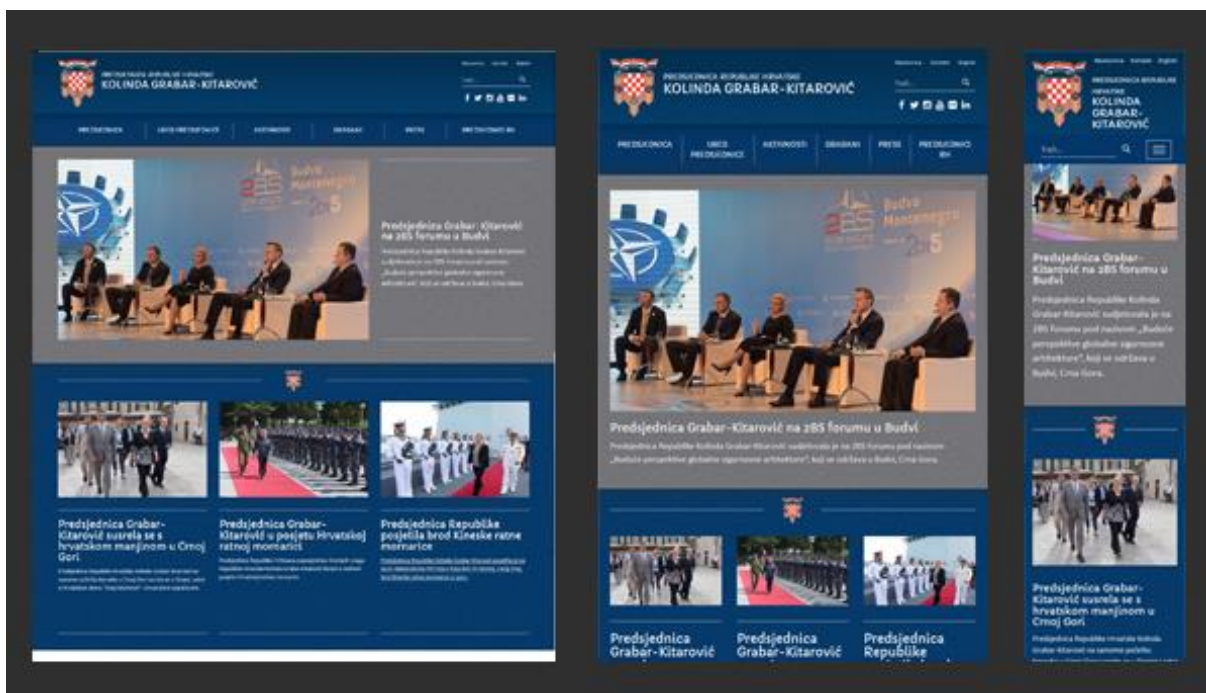
Bootstrap je *framework* otvorena koda stvoren za internu uporabu u Twitteru, no koji je zbog svoje jednostavnosti i pristupačnosti ubrzo ušao u širu uporabu. Prije no što je bio otvorena koda zvao se *Twitter Blueprint* i bio je dio Twittera. Nakon što je Twitter održao svoj prvi *Hack Week* događaj gdje se potiče programere na inovativnost, projekt je postao iznimno popularan. Na kraju je služio kao stilski vodič za tvrtku preko godinu dana prije no što je pušten u javnost 19.10.2011. godine. (Otto, Thornton, 2011.)

Tek u verziji 3 *Bootstrap* je ponovno napisan s *Mobile First* pristupom u vidu i tada je u njega implementiran prilagodljivi dizajn. *Mobile First* pristup je prvi definirao Wroblewski (Wroblewski, 2009.) koji kaže da se web stranice rade prvo za mobilne uređaje zbog toga što njihove niske rezolucije i male dijagonale ekrana ne dopuštaju zatrpavanje korisnika sadržajem. Stranica se prvo dizajnira i skalira za mobilne uređaje, a zatim se prilagođava prikazu na *desktop* računalima kao što je prikazano na slici 10.



Slika 10 Kako funkcionira dizajn web stranice s *Mobile First* pristupom⁸

Framework Bootstrap je sada na vrhuncu popularnosti i njegov utjecaj se može vidjeti i na našem području (Vidi sliku 11).



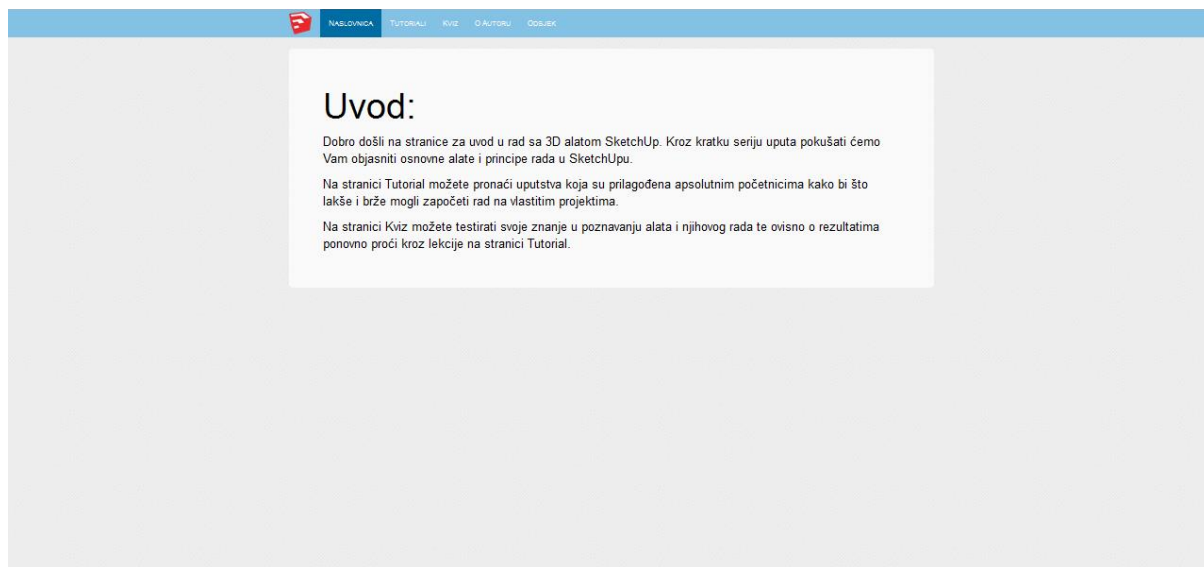
Slika 11 Prilagodljivi dizajn u Bootstrapu vidljiv na stranici hrvatske predsjednice⁹

U svojoj srži, Bootstrap pristupa prilagodljivom dizajnu putem sustava od maksimalno 12 stupaca i mreže široke 940 piksela. Stupci se mogu podijeliti na nekoliko načina i time pridonijeti kvaliteti prikaza stranice. Za vrijeme izgradnje stranice krovni se elementi mogu definirati ili opisati s vrijednostima *fluid* ili *fixed-width*. Ako se radi o fluidnome elementu, on

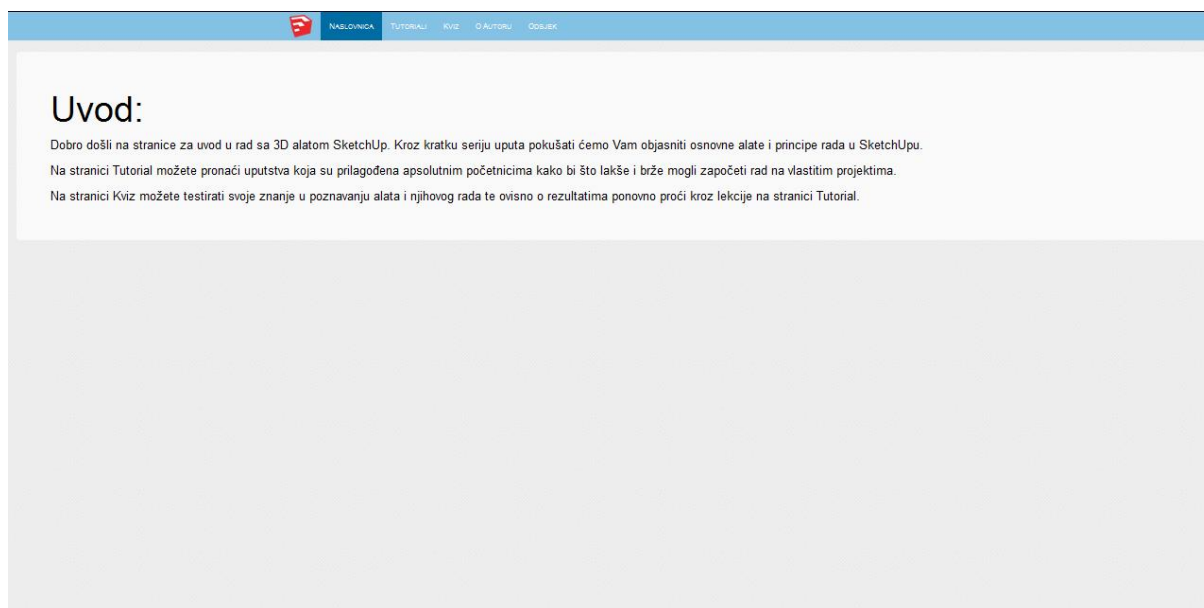
⁸ Preuzeto sa: <http://zurb.com/word/mobile-first>

⁹ Preuzeto sa: <http://predsjednica.hr/>

će se maksimalno proširiti, dok će *fixed-width* element koristiti zadanu širinu, kao što je prikazano na slikama 12 i 13.



Slika 12 Naslovnica u Bootstrapu uz spremnik fiksne veličine



Slika 13 Naslovnica u Bootstrapu uz spremnik fluidne veličine

Kao što je vidljivo iz html koda u kodnom bloku 4, jedina razlika je što se pri pozivanju div elementa kao klasa navodi „container-fluid“ u slučaju ako želimo fluidni element (s desna), a

ako želimo fiksni navodimo samo klasu „container“ (s lijeva) koju zatim naknadno oblikujemo u CSS dokumentu.

<pre> <div class="container"> <div class="jumbotron"> <h1>Uvod:</h1> <p>...</p> </div> </div> </pre>	<pre> <div class="container-fluid"> <div class="jumbotron"> <h1>Uvod:</h1> <p>...</p> </div> </div> </pre>
--	--

Kodni blok 4 Usporedba fixed (lijevo) i fluid (desno) elemenata

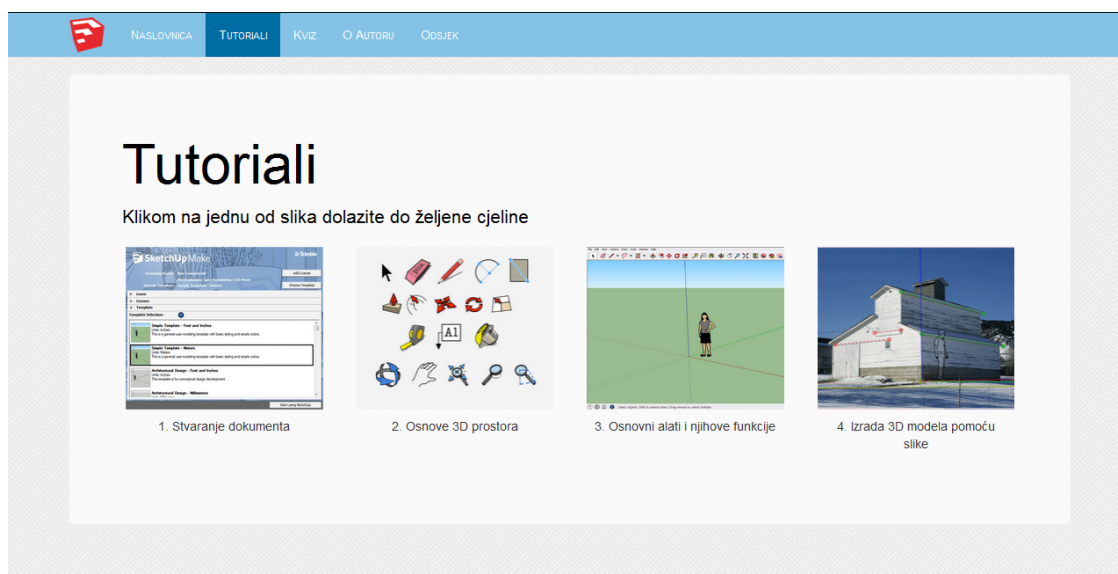
Što se same podjele stupaca tiče, najbolje je referirati se direktno na tablicu na Bootstrapovoj stranici prikazanoj na slici 14. Jasno je istaknuto da su stupci podijeljeni u 4 kategorije ovisno o razlučivosti ekrana uređaja koji im pristupa.

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

Slika 14 Tablica preuzeta s Bootstrapove stranice, koja prikazuje na koji način su podijeljene klase za prikaz stupaca

Kao praktičan primjer funkcioniranja prikaza stupca možemo uzeti poveznicu Tutoriali na mojoj stranici. Sljedeći primjer će koristiti najosnovniji prikaz slika u stupcima putem „col-md-3“ elementa koji je prema tablici namijenjen ekranima razlučivosti većim od 992 piksela i uzevši u obzir broj 3 na kraju imena element će svojom veličinom zauzimati 3 stupca od 12

odnosno moguće je prikazati 4 elementa te veličine po retku. Primjer toga možemo vidjeti na slici 14.



Slika 15 Stranica Tutoriali na kojoj je prikazan primjer 4 elementa širine 3 stupca za srednju razlučivost ekrana

```
<div class="row">
  <div class="col-md-3">
    <div class="thumbnail">
      <a href="tutorial-1.html">
        
      </a>
      <div class="caption">
        1. Stvaranje dokumenta
      </div>
    </div>
    ...
  </div>
```

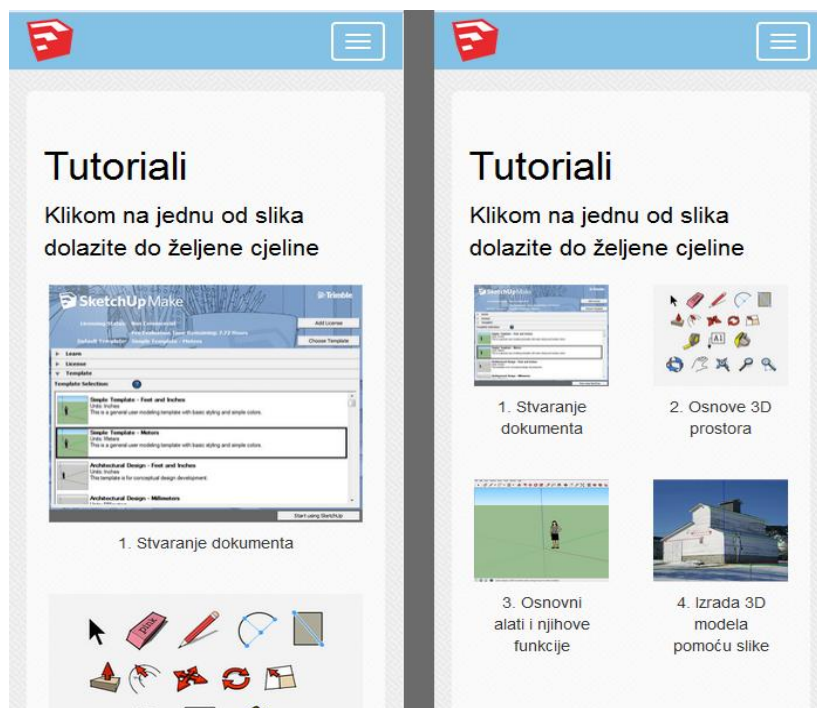
Kodni blok 5 Izvadak iz html-a gdje se prikazuje pozivanje klase za srednju razlučivost širine 3 stupca

Stoga kao što je prikazano u kodnom bloku 5 potrebno je otvoriti element `<div>` klase „row“ i zatim još jedan u kojem će biti sadržaj na čiji prikaz želimo utjecati te klasu nazovemo „col-md-3“ da bi to Bootstrap mogao prikazati na zadani način.

Elementi su na ekranu uredno prikazani, no prilagođeni su samo srednjim razlučivostima. Budući da stranicu treba prilagoditi i nižim i višim razlučivostima kako elementi ne bi bili preveliki ili presitni, u ime klase dodaje se „xs“ ili „lg“. Time se dolazi do mehanizama za prikaz na vrlo visokim i niskim razlučivostima.

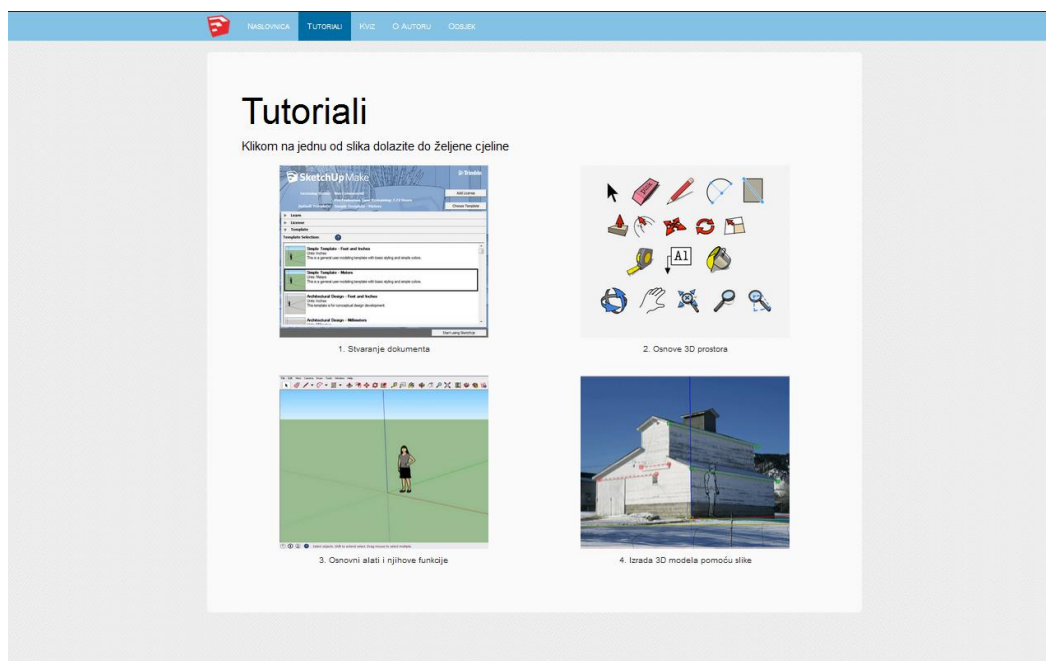
```
<div class="row">
  <div class="col-xs-6 col-md-3 col-lg-6">
    <div class="thumbnail">
      <a href="tutorial-1.html">
        
      </a>
      <div class="caption">
        1. Stvaranje dokumenta
      </div>
    </div>
    ...
  </div>
```

Kodni blok 6 Nadopunjavanje imena klase sa podacima za niske i visoke razlučivosti



Slika 16 Usporedba stranice bez (lijevo) i sa (desno) posebno podešenih naziva za male razlučivosti

Kao što je vidljivo na slici 16, na mobilnim uređajima puno bolje izgleda desno rješenje gdje je korišten „col-xs-6“ naziv. On oblikuje sadržaj na način da u slučaju malenog ekrana zauzima 6 od mogućih 12 stupaca čime se zadržava preglednost stranice.



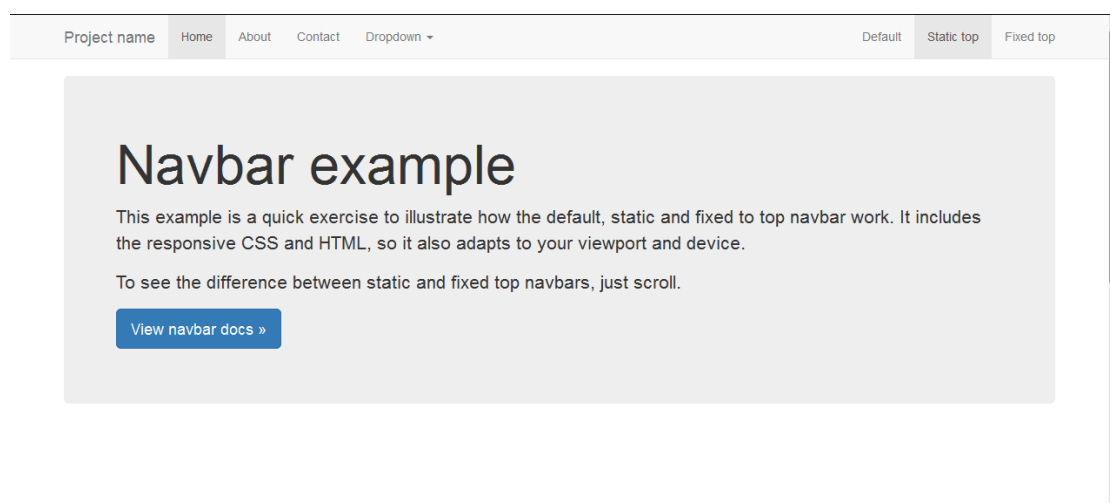
Slika 17 Prikaz na visokoj razlučivosti uz naziv klase "col-lg-6"

Priložena slika 17 pokazuje da je i na visokoj razlučivosti prikaz pregledniji pri korištenju naziva „col-lg-6“ nego korištenje naziva klase „col-md-3“, koja bi na višim razlučivostima slike ostavila malenima.

Još je jedna od velikih prednosti Bootstrapa mogućnost skidanja već gotovih predložaka s Bootstrapove *Getting started* stranice (<http://getbootstrap.com/getting-started/>), a koje se može prilagoditi vlastitim potrebama.

To neće učiniti izgled stranice originalnim, no uvelike ubrzava proces njene izrade. Za svoju sam stranicu koristio predložak „Static Top Navbar Example“, prikazan na slici 18.

Predložak sam preuzeo i prilagodio koristeći velik dio CSS-a koji sam već napisao za prethodne iteracije stranice kako bi vizualni stil ostao konzistentan. Trebalo mi je puno manje vremena da postignem željeni rezultat i imam spremnu mobilnu verziju stranica.



Slika 18 Predložak

3.2.2. Ostali frameworkci

Bootstrap nije jedini framework za izradu prilagodljivih web stranica. U ovom ću radu spomenuti još neke. Prema Agus (Agus, 2015.) popularni su još:

1. Skeleton
2. BassCSS

3. CardinalCSS
4. Base
5. ConciseCSS

Svi navedeni *frameworki* su dizajnirani s idejom *Mobile First* i u sebi sadrže sustav stupaca koji se dinamično prilagođava širini prozora preglednika.

Neki od njih poput *Base* su vrlo minimalistički i ne sadrže velik spektar dodatnih mogućnosti, već samo stupčani sustav i neke od najosnovnijih klasa za oblikovanje. *Base* je dovoljno malen i jednostavan da funkcionira čak i na Internet Exploreru 7 koji je izdan 2006. godine i ima iznimno lošu CSS podršku (Agus, 2015.).

Osobno sam se odlučio za *framework* Bootstrap zbog toga što je u trenutku kada sam krenuo raditi na vlastitoj web stranici bio najbolje dokumentiran i sadržavao je najviše primjera. Naposljetku se to pokazalo kao dobar izbor jer je sam *framework* sveobuhvatan i nije bilo potrebno pretraživati Internet u potrazi za implementacijom nekih dodatnih mogućnosti jer je sve unaprijed implementirano.

4. Zaključak

U ovom radu sam se pokušao ukratko dotaknuti početaka mobilnog weba kako bih postavio temelj za razvijanje ostatka rada.

U radu sam opisao svoj put kroz izradu rješenja za prikaz web stranica na mobilnim uređajima, s ciljem demonstracije problema s kojima sam se i sam susreo. Krenuvši od preusmjeravanja korisnika na mobilnim uređajima pomoću JavaScripte i PHP-a, da bi od toga odustao zbog previše iznimaka i problema. Dotaknuo sam se *@media* upita uvedenih u CSS verziji 3 koji su bili na tragu nečeg velikog, da bi na kraju sve kulminiralo prilagodljivim web dizajnom uz pomoć *frameworka* Bootstrap.

Prošavši kroz povijest mobilnog weba i susrevši se sa nekim dijelovima te iste povijesti u vlastitom kodu dobio sam puno bolji uvid u mobilni web nego što sam ga imao prije. WAP tehnologija je tehnologija kojoj se sasvim sigurno ne bi trebali vraćati dok je preusmjeravanje kao pristup zastarjelo iako još uvijek ima svoju publiku.

Osobnog sam mišljenja da treba prihvatiti glavnu ideju W3C-a u objedinjavanju cijelog weba, a ne daljnjem cjeplkanju na platforme i uređaje. Prilagodljivi web dizajn je svakako korak u tom smjeru i mišljenja sam da će se u sljedećih nekoliko godina još više prilagoditi i proširiti.

5. Popis literature

1. Agus, "**10 Lightweight Alternatives To Bootstrap & Foundation**",
<http://www.hongkiat.com/blog/bootstrap-alternatives/>, (posjećeno: 5.6.2015.), 2015.
2. Beal, Vangie, "**WAP - Wireless Application Protocol**",
<http://www.webopedia.com/TERM/W/WAP.html>, (posjećeno: 15.6.2015.), 2012.
3. Dhawan, Chander, "**i-mode: What is i-mode**",
http://www.mobileinfo.com/imode/what_is.htm, (posjećeno: 24.6.2015.), 2001a.
4. Dhawan, Chander, "**WAP- What is Wap?**",
http://www.mobileinfo.com/wap/what_is.htm, (posjećeno: 23.6.2015.), 2001b.
5. Everts, Tammy, "**Six reasons to ditch your .m site in 2014.**",
<http://www.webperformancetoday.com/2013/12/04/six-reasons-to-ditch-your-m-site-in-2014>, (posjećeno: 22.5.2015.), 2013.
6. Firtman, Maximiliano R. "**Programming the Mobile Web. 2nd ed.**", Sebastopol, CA: O'Reilly Media, 2013.
7. Gasston, Peter. "**The Modern Web, Multi-device web development with html5, css3, and javascript**" S.l. No Starch Press, 2013.
8. Ghazarian, Armen, "**The Pros and Cons of Responsive Web Design vs. Mobile Website vs. Native App**", <http://designmodo.com/responsive-design-vs-mobile-website-vs-app/>, (posjećeno: 2.6.2015.), 2014.
9. Greenfield, Daniel, Audrey, Roy, "**Django Packages – Mobile**",
<https://www.djangopackages.com/grids/g/mobile/>, (posjećeno: 21.5.2015.), 2013.
10. Hay, Stephen. "**There Is No Mobile Web.**", <http://www.the-haystack.com/2011/01/07/there-is-no-mobile-web/>, (posjećeno: 14.6.2015.), 2011.
11. Johnston, Mike „**CMS or WCM – Which is Which?**“ <http://www.cmscritic.com/cms-or-wcm-which-is-which/>, (posjećeno: 28.6.2015.), 2011.
12. Markle, Brad, "**How to redirect your website to its mobile version**",
<http://www.inmotionhosting.com/support/website/redirects/mobile-redirect>,
(posjećeno: 23.5.2015.), 2015.
13. Mills, Crhis, "**Introduction to Mobile Web.**",
http://www.w3.org/wiki/Introduction_to_mobile_web, (posjećeno: 14.6.2015.), 2011.

14. Murtaugh, Rebecca. "**Mobile Now Exceeds PC: The Biggest Shift Since the Internet Began**", <http://searchenginewatch.com/sew/opinion/2353616/mobile-now-exceeds-pc-the-biggest-shift-since-the-internet-began>, (posjećeno: 15.06.2015.), 2014.
15. NTTDoCoMo, "**i-mode**", <https://www.nttdocomo.co.jp/english/service/imode/>, (posjećeno: 23.6.2015.), 2010.
16. NTTDoCoMo, „**NTT DoCoMo and Cellcom Israel Commence Strategic Partnership for i-mode in Israel**“, https://www.nttdocomo.co.jp/english/info/media_center/pr/2004/001216.html, (posjećeno: 27.6.2015.), 2004.
17. Otto, Mark, Thornton, Jacob, „**About Bootstrap**“, <http://getbootstrap.com/about/>, (pristupljeno 28.6.2015.), 2011.
18. Sandu, Bogdan, "**Detecting and Redirecting Mobile Users**", <http://www.designyourway.net/blog/resources/detecting-and-redirecting-mobile-users>, (posjećeno: 30.5.2015.), 2011.
19. Spurlock, Jake, "**Bootstrap**", O'reillymedia, 2013.
20. Vodafone, "**A brief history of mobile data: The road to the next generation...**", blog.vodafone.co.uk/2013/01/21/a-brief-history-of-mobile-data/, (posjećeno: 22.6.2015.), 2013.
21. W3Schools, "**W3Schools**", <http://www.w3schools.com>, (posjećeno: 23.5.2015.)
22. Wilson, Raelene, "**8 Effective Plugins for Optimizing Your Word Press Site for Mobile**", <http://premium.wpmudev.org/blog/plugins-optimizing-wordpress-site-mobile/>, (posjećeno: 21.5.2015.), 2014.
23. Wroblewski, Luke, „**Mobile First**“, <http://www.lukew.com/ff/entry.asp?933>, (posjećeno: 28.6.2015.), 2009.